IN THE CLAIMS:

Please amend Claims 1, 21, 33 and 41 as follows:

1. (Currently Amended) A method of searching for a computer address in an address table, the computer address having a bit size n, the steps comprising:

partitioning the bit size n computer address into an upper set of n-m bits and a lower set of m bits, wherein m comprises a bit size less than bit size n;

generating a search index by compressing the upper set of n-m bits to obtain a compressed value of the computer address, wherein the search index comprises a number of bits equal to the number of bits of the lower set of m bits;

accessing a primary address record corresponding to the computer address in a primary address table, the primary address record being accessed by using the search index to locate the primary address record, wherein the primary address record includes the computer address, a port number associated with the computer address, and a link that ~~that~~ specifies the location of an initial secondary address record in a secondary address table;

comparing the search index to the primary address record by:

selecting the m low order bits of the combination of the search index and the lower set of m bits, wherein a first value is determined,

decompressing the compressed value of the address contained in the primary address record to obtain a second value, and

comparing the first value to the second value; and

if the first value does not equal the second value, then accessing the initial secondary address record using the link, wherein the initial secondary address record includes a respective address entry of the bit size n-m, a port number associated with the computer address, and a link to a subsequent secondary address record of the same hash family.

2. (Original) The method of claim 1, wherein the secondary address table is stored in content addressable memory.

3. (Original) The method of claim 1, wherein the primary address table is

stored in a memory external to the switch.

4. (Original) The method of claim 1, wherein the step of generating a search index by compressing the computer address further comprises compressing the computer address from a width of 48 bits to a width of less than 48 bits.

5. (Original) The method of claim 1, wherein the step of generating a search index by compressing the computer address further comprises compressing the computer address from a width of 48 bits to a width of 16 bits.

6. (Original) The method of claim 1, further comprising the step of comparing the first value to the computer address in the secondary record.

7. (Original) The method of claim 6, further comprising, if the subsequent secondary record is empty, the step of populating the subsequent secondary record with the computer address and with a port associated with the computer address.

8. (Original) The method of claim 7, further comprising, if the subsequent secondary record is empty, the step of populating the initial secondary address record with the location of the subsequent secondary address record.

9. (Original) The method of claim 1, wherein the bit size n computer address comprises 60 bits, the upper set of n-m bits comprises 48 bits and the lower set of m bits comprises 12 bits.

10. (Original) The method of claim 9, wherein the upper set of n-m bits is a MAC address.

11. (Original) The method of claim 9, wherein the lower set of m bits is a VLAN identifier.

12. (Currently Amended) A storage and search unit for computer addresses each having a fixed bit size n, the unit comprising:

a primary address table stored within a first memory of a first bus width, the primary address table configured to store a plurality of primary address records, each primary address record including a respective address entry of a first bit size less than the fixed bit size n, a port number associated with the compressed address entry and a first link that links each primary address record to a corresponding chain of secondary address records in a ~~second~~ secondary address table;

a secondary address table stored within a second memory separate from the first memory, the ~~second~~ secondary address table configured to store a plurality of secondary address records, each secondary address record including a respective address entry of the first bit size less than the fixed bit size n, a port number associated with the computer address, and a link that links each secondary address record to a corresponding secondary address record in the ~~second~~ secondary address table to thereby form one or more linked chains of secondary address records, wherein each chain of secondary address records contains full address entries of the same hash family;

a software search module configured to store and access the primary address records and secondary address records, wherein the software module stores each primary address record in a location defined by the value of the respective compressed address entry.

13. (Original) The storage and search unit of claim 12, wherein the computer addresses comprise MAC addresses.

14. (Original) The storage and search unit of claim 12, wherein the computer addresses comprise Internet Protocol addresses.

15. (Original) The storage and search unit of claim 12, wherein the second memory comprises a content addressable memory.

16. (Original) The storage and search unit of claim 12, wherein the bus width of the first memory is 16 bits.

17. (Original) The storage and search unit of claim 12, wherein the bit size of the compressed address entry is equal to the bus width of the first memory.

18. (Original) The storage and search unit of claim 12, wherein the storage and search unit comprises a switch on an Ethernet network.

19. (Original) The storage and search unit of claim 18, wherein the first memory is external to the switch.

20. (Original) The storage and search unit of claim 19, wherein the second memory is internal to the switch.

21. (Currently Amended) Computer readable software stored within a frame forwarding device of a computer network, the computer readable software code including

a set of instructions for causing the device to search for a computer address in an address table, the computer address having a bit size n, the instruction further causing the device to:

partition the bit size n computer address into an upper set of n-m bits and a lower set of m bits, wherein m comprises a bit size less than bit size n;

generate a search index by compressing the upper set of n-m bits to obtain a compressed value of the computer address, wherein the search index comprises a number of bits equal to the number of bits of the lower set of bits;

access a primary address record corresponding to the computer address in a primary address table, the primary address record being accessed by using the search index to locate the primary address record, wherein the primary address record includes the computer address, a port number associated with the computer address, and a link ~~that~~ that specifies the location of an initial secondary address record table;

compare the search index to the primary address record by:

selecting the m low order bits of the combination of the search index and the lower set of m bits, wherein a first value is determined,

decompressing the compressed value of the address contained in the primary address record to obtain a second value, and

comparing the first value to the second value; and

if the first value does not equal the second value, then access the initial secondary address record using the link, wherein the initial secondary address record includes a respective address entry of the first bit size less than the fixed bit size n, a port number associated with the computer address, and a link to a subsequent secondary address record of the same hash family.

22. (Original) The computer readable software code of claim 21, wherein the primary address table is stored in a memory external to the switch.

23. (Original) The computer readable software code of claim 21, wherein the secondary address table is stored in a content addressable memory.

24. (Original) The computer readable software code of claim 21, further including instructions for causing the device to compress the computer address from a

width of 48 bits to a width of less than 48 bits.

25. (Original) The computer readable software code of claim 21, further including instructions for causing the device to compress the computer address from a width of 48 bits to a width of 16 bits.

26. (Original) The computer readable software code of claim 21, further including instructions for causing the device to compare the first value to the full computer address in the secondary record.

27. (Original) The computer readable software code of claim 26, further including instructions for causing the device to, if the subsequent secondary record is empty to populate the subsequent secondary record with the computer address and with a port associated with the computer address.

28. (Original) The computer readable software code of claim 27, further including instructions for causing the device to, if the subsequent secondary record is empty, populate the initial secondary address record with the location of the subsequent secondary address record.

29. (Original) A method of searching for a computer address in an address table, the computer address having a bit size n, the steps comprising:

partitioning the bit size n computer address into an upper set of n-m bits and a lower set of m bits, wherein m comprises a bit size less than bit size n;

generating a search index by compressing the upper set of n-m bits to obtain a compressed value of the computer address, wherein the search index comprises a number of bits equal to the number of bits of the lower set of bits;

accessing an address record corresponding to the computer address in an address table, the address record being accessed by using the search index to locate the address record, wherein the address record includes the computer address, and a port number associated with the computer address; and

comparing the search index to the address record by:

selecting the m low order bits of the combination of the search index and the lower set of m bits, wherein a first value is determined,

decompressing the compressed value of the address contained in

the address record to obtain a second value, and

comparing the first value to the second value.

30. (Original) The method of claim 29, wherein the address table is stored in a memory external to the switch.

31. (Original) The method of claim 29, wherein the step of generating a search index by compressing the computer address further comprises compressing the computer address from a width of 48 bits to a width of less than 48 bits.

32. (Original) The method of claim 29, wherein the step of generating a search index by compressing the computer address further comprises compressing the computer address from a width of 48 bits to a width of 16 bits.

33. (Original) A method of searching for a computer address in an address table, the computer address having a bit size n, the steps comprising:

generating a search index by compressing the computer address to obtain a compressed value of the address, wherein the search index comprises a first number of bits less than the bit size n;

accessing at least two primary address records corresponding to an equal number of computer addresses stored in a primary address table, the at least two primary address records being accessed by using the search index to locate the primary address records,

wherein the primary address records include the computer addresses, a port number associated with each of the computer addresses, and a link that specifies the location of an initial secondary address record in a secondary address table;

comparing the search index to the primary address records simultaneously by:

decompressing the search index to obtain a first value,

decompressing the compressed values of the addresses contained in each of the primary address records, and

comparing the first value to the values of the addresses of the primary address records; and

if the first value does not equal any of the values of the addresses of the

primary address records, then accessing the initial secondary address record using the link, wherein the initial secondary address record includes a computer address, a port number associated with the computer address, and a link to a subsequent secondary address record of the same hash family.

34. (Original) The method of claim 33, wherein the secondary address table is stored in content addressable memory.

35. (Original) The method of claim 33, wherein the primary address table is stored in a memory external to the switch.

36. (Original) The method of claim 33, wherein the step of generating a search index by compressing the computer address further comprises compressing the computer address from a width of 48 bits to a width of less than 48 bits.

37. (Original) The method of claim 33, wherein the step of generating a search index by compressing the computer address further comprises compressing the computer address from a width of 48 bits to a width of 16 bits.

38. (Original) The method of claim 33, further comprising the step of comparing the first value to the computer address in the secondary record.

39. (Original) The method of claim 38, further comprising, if the subsequent secondary record is empty, the step of populating the subsequent secondary record with the computer address and with a port associated with the computer address.

40. (Original) The method of claim 39, further comprising, if the subsequent secondary record is empty, the step of populating the initial secondary address record with the location of the subsequent secondary record.

41. (Currently Amended) A method for forwarding a frame to a computer address using classification based upon multiple fields in a header, the header having a first field of bit size n-m and a second field having a bit size m, the steps comprising:

concatenating the first field of bit size n-m and the second field of bit size m into a bit size n, wherein m comprises a bit size less than bit size n;

generating a search index by compressing the concatenated bit size n to obtain a compressed value of the concatenated fields, wherein the search index comprises a number of bits equal to the number of bits of the second field bit size m;

accessing a primary record corresponding to the computer address in a primary table, the primary record being accessed by using the search index to locate the primary record, wherein the primary record includes the computer address, a port number associated with the computer address, and a link ~~that~~ that specifies the location of an initial secondary record in a secondary table;

comparing the search index to the primary record by:

selecting the concatenated fields of bit size m of the search index and the second field of bit size m, wherein a first value is determined,

decompressing the compressed value of the address contained in the primary record to obtain a second value, and

comparing the first value to the second value; and

if the first value does not equal the second value, then accessing the initial secondary record using the link, wherein the initial secondary record includes a respective computer address entry of the bit size n-m, a port number associated with the computer address, and a link to a subsequent secondary record of the same hash family.

42. (Original) The method of claim 41, wherein the concatenated fields of bit size n comprise 60 bits, the first field of n-m bits comprises 48 bits and the second field of m bits comprises 12 bits.

43. (Original) The method of claim 42, wherein the first field of n-m bits is a MAC address.

44. (Original) The method of claim 42, wherein the second field of m bits is a VLAN identifier.

45. (Original) The method of claim 41, wherein the secondary table is stored in content addressable memory.

46. (Original) The method of claim 41, wherein the primary table is stored in a memory external to the switch.

47. (Original) The method of claim 41, wherein the step of generating a search index by compressing the concatenated first and second fields further comprises compressing the concatenated fields from a width of 60 bits to a width of less than 60 bits.

48. (Original) The method of claim 41, wherein the step of generating a search index by compressing the concatenated first and second fields further comprises compressing the concatenated fields from a width of 60 bits to a width of 12 bits.

49. (Original) The method of claim 41, further comprising the step of comparing the first field to the computer address in the secondary record.

50. (Original) The method of claim 49, further comprising, if the subsequent secondary record is empty, the step of populating the subsequent secondary record with the computer address and with a port associated with the computer address.

51. (Original) The method of claim 50, further comprising, if the subsequent secondary record is empty, the step of populating the initial secondary record with the location of the subsequent secondary record.

52. (Original) The method of claim 41, further comprising the step of compressing the first field of bit size n-m and the second field of bit size m.

53. (Original) The method of claim 52, further comprising the step of concatenating the compressed first field of bit size n-m and the compressed second field of bit size m into a bit size n, wherein m comprises a bit size less than bit size n.

54. (Original) The method of claim 53, wherein the step of generating a search index further comprises compressing the concatenated compressed fields of bit size n to obtain the compressed value of the concatenated fields.